



# [12] 发明专利申请公开说明书

[21] 申请号 01805185.5

[43] 公开日 2004年12月1日

[11] 公开号 CN 1552018A

[22] 申请日 2001.12.10 [21] 申请号 01805185.5

[30] 优先权

[32] 2000.12.18 [33] US [31] 09/739,517

[86] 国际申请 PCT/IB2001/002431 2001.12.10

[87] 国际公布 WO2002/050674 英 2002.6.27

[85] 进入国家阶段日期 2002.8.16

[71] 申请人 皇家飞利浦电子有限公司

地址 荷兰艾恩德霍芬

[72] 发明人 J·W·路易斯

[74] 专利代理机构 中国专利代理(香港)有限公司

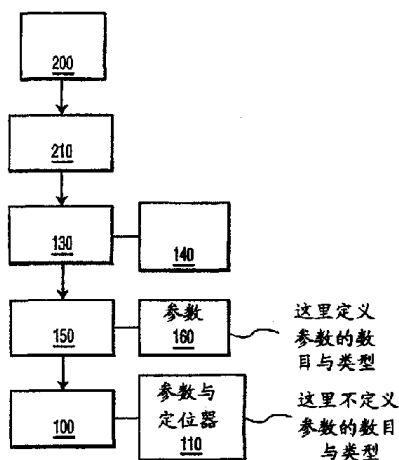
代理人 吴立明 王 勇

权利要求书1页 说明书5页 附图3页

[54] 发明名称 软件编译器的嵌入系统的坚固记录系统

[57] 摘要

预处理脚本分析记录语句的消息目录。例如，各记录可包含不同的记录消息、格式串、及变量的定位器加说明。脚本然后生成定义包含在消息目录中的各种类型的消息的首标文件。然后它为各种类型的消息定义宏。当程序员编写代码时，他/她使用宏格式而不是标准语言格式。编译时这些宏归结为对通用记录调用的具体实例所需的相应参数的类型与数目的函数的调用。在最终编译代码时，当参数的数目与类型不匹配时，编译器将生成出错消息。



1. 一种编程操作的方法，包括下述步骤：  
定义包含最终执行记录操作的语句的函数（150）；  
所述函数包含对特定的记录操作专用的至少一个数目与类型的至少一个参数（160）；  
5 所述语句是对数目与类型的至少一个的参数通用的；  
在程序中调用所述函数，借此通过编译所述程序便可揭示对所述特定记录操作专用的数目与类型的所述至少一个之间的不匹配。
2. 权利要求1中的方法，其中所述操作为记录操作。
- 10 3. 一种带有包含下述各项的程序的软件介质：  
带有在运行时间最终执行记录操作的语句的函数定义；  
所述函数包含对特定记录操作专用的数目与类型的至少一个的至少一个参数；  
所述语句是对数目与类型的至少一个的参数通用的；  
15 所述程序调用所述函数，借此通过编译所述程序可以揭示对所述特定记录操作专用的数目与类型的所述至少一个之间的不匹配。
4. 权利要求3中的软件介质，其中所述操作为记录操作。
5. 一种编程记录脚本的方法，包括下述步骤：  
定义相应的参数的数目与/或类型的函数，所述参数是传递给要求  
20 至少所述参数的程序语句的；  
生成包含记录消息的类别的记录消息目录；  
定义宏翻译编程语句到对所述消息目录起作用的所述函数的各自的一个中；  
在程序中引用所述宏。
- 25 6. 权利要求5中的方法，其中所述定义宏的步骤包含定义有条件地将文件名与/或行号附加在各自的记录消息上的宏。

## 软件编译器的嵌入系统的坚固记录系统

## 发明背景

## 5 发明领域

本发明涉及用于定义计算机软件中的记录操作的机构而更具体地涉及用于记录需要固定数目与/或类型的参数的语句的这种机构,这种机构是诸如导致在编译时间检验参数的数目与类型的。

## 背景技术

10 记录是用来描述出现在进行中的软件进程中的检查事件的所有方式的通用术语。熟悉的实例为在通过调制解调器连接到计算机上时步骤的跟踪。随着连接中各步骤的完成,记录系统输出消息给连接终端。如果出现错误,便立即有可能确定在该错误停止它之前进程已进行多远。通常记录导致生成下一消息,能任选地以压缩标记形式存储消息  
15 直到以后用可读的正文替换这些标记时读取它们。

软件通常是编写成使得相同的代码能被不同地点中说不同语言的用户使用的。不是将备择的检查语句嵌入可执行软件本身中,而是用消息目录来存储备择的语言格式。将软件编写成只生成检查语句的代码或标准形式,然后可用目录转换它们。转换可在读取记录输出时进  
20 行(假定标准输出是临时存储的)或者在生成标准形式之后立即转换并在输出或存储装置上输出它。

在一些系统中,用来生成检查输出的语句可采用固定数目与通用类型的参数。例如,语句可采用格式化串及诸如整数、串、浮点值等各式各样类型中任何参数的固定长度系列。这些语句可以是对操作系统专用的或者是程序语言的一部分。如果参数的数目是固定的,例如,  
25 不是所有都可使用的。如果参数可以是任何类型的,即使特定格式语句与它们不一致,也难以保证已准确地编程了这些调用。这是因为在编译期间,不检验这些参数的类型与/或数目对所定义的特定记录事件是否适当。唯一的选择是通过执行来测试记录语句,但这是费事且通常不现实的。最终,运行时间测试并不提供何处出现记录错误的方便  
30 指示。较新的面向对象(OO)的语言提供一种解决方法,但在诸如嵌入系统等一些系统中,可将软件作者的至少某些代码部分限制在非OO

语言上。

### 发明概述

5 本发明通过提供不是对语言而是对各种类型的记录语句的依赖于消息目录的预处理脚本 (script) 来解决先有技术难于避免的上述问题。该消息目录包含各定义在一个记录中的消息结构。例如, 各记录可包含不同的记录消息、格式串、及变量的占位器加说明。当编译器的记录语句采用通用语句、后面跟随格式语句与固定数目参数的形式时, 后者是适当的。

10 本发明使用带预处理脚本的消息目录, 前者分析消息目录来确定各记录所需的参数数目。然后脚本生成为消息目录中所定义的各种消息定义宏的首标文件。程序员在编写代码时使用宏格式而不是标准语言格式。编译时宏分解成对包含标准语言格式中的调用的函数的调用。然而, 该函数是对特定消息所需的类型与/或参数数专用的。在最终编译代码时, 预处理器已用相应数目与/或类型的参数的函数调用替换了所有的宏。从而在这一情况中, 当参数的数目与/或类型不匹配时, 编译器将生成出错消息。

15 下面参照示例性图结合一定较佳实施例描述本发明, 以便能更全面地理解它。参照附图, 强调所示出的详细描述是用示例方式的并且只是为了本发明的较佳实施例的说明性讨论的目的, 并且是为了提供认为是本发明的原理与概念方面的最有用与容易理解的描述而提出的。这一方面, 并不试图示出比对本发明的基本理解所必需的更详细的发明的结构细节, 结合附图作出的描述使熟悉本技术的人员清楚在实践中如何实施本发明的若干形式。

### 附图描述

25 图 1 为按照本发明的实施例的用于检验记录调用错误的进程的流程图。

图 2 为表示构成记录调用的先有技术的框图。

图 3 为按照本发明的实施例构成记录调用的方法的框图。

30 图 4 为说明按照本发明的实施例使用宏的自动生成来提供可在编译时揭示的诊断信息的构成记录调用的方法的框图。

参见图 1, 程序员通过增加生成新记录输出的调用定期编写新源码或更新老源码。如果这些调用具有以前不包含在老代码中的类型, 或

者如果正在起草新程序，则存在新的消息类型（S10）并在步骤 S20 中生成新的或更新的消息目录。消息目录是记录消息类型的表。在本发明的实施例中，消息目录具有下表中所示的格式。

Rec.	Name	Format	F1	F2	F3 ...	Fn	Descr.

5

在步骤 S25 中定义采用各种类型的消息所需的参数的数目与/或类型的函数。这些函数可以是对一类记录消息类型通用的。例如，对于一个参数的消息可有相应的函数，对于需要两个参数的消息可有另一函数，等。作为替代，对于参数的数目与类型的各种组合可具有独立的函数。

10

在步骤 S30 中，分析消息目录并生成为消息目录中的各种类型的记录消息建立宏定义的首标（即，每一记录一个）。各宏是定义成调用适当函数的。在步骤 S40 中将得出的新首标文件加在新源码文件上或替换老的首标文件，并将基于新消息类型的任何新源码加在源码上。然后在步骤 S50 中运行编译器并在步骤 S60 中检查由参数数目或类型不匹配导致的任何错误及采取适当的纠正行动。

15

C 文件的一部分中的首标的示例如下。

```

/* Log message indexes */
#define LoggingVersion      0x0001
#define VideoSyncLost      0x0002
#define OutputLocked       0x0004
/* Log call macros */
#define LOG_LoggingVersion(v0) (vlog1Event(LoggingVersion, (v0)))
#define LOG_VideoSyncLost() (vlog0Event(VideoSyncLost))
#define LOG_OutputLocked(v0, v1, v2) (vlog3Event(OutputLocked, (v0),(v1),(v2)))

```

20 然后程序员便能使用其代码中的记录调用宏。例如，可出现下述语句。

```
LOG_OutputLocked(2, 0, 1)
```

在编译期间它被下述语句替换

```
(vlog3Event(0x0004, (2),(0),(1)))
```

从而，当编译器在预编译步骤中遇到这些语句之一时，它使用在记录调用宏定义中所定义的函数调用替换该宏正文。换言之，LOG\_LoggingVersion(XYZ)的每一次出现都将改变成vlog1Event(0xNNNN, XYZ)。其中

5 “0xNNNN”为特定记录事件的消息号而“XYZ”为宏语句中使用的参数。

注意这里的要点在于在记录调用宏定义中根据参数的数目使用不同的函数。这些函数实际上执行生成记录输出所需的步骤。各函数包含适当的记录语句，但和在先有技术中一样在函数定义内，该或这些语句在参数的数目与类型方面是相同的。通过将参数传递过与所要求的参数的数目与类型相应的函数，便将相应的函数束缚在宏定义中所要求的记录事件的类型上，使得编译器在编译期间能检验宏语句中的参数的数目与类型。这在运行时间之前得出出现不正确的语法的精确位置的指示。

10

下面是函数定义的示例清单。下面的代码定义采用四个参数的记录函数。

15

```
{
    logMsg(format[msgNumber], (int)v0, (int)v1, (int)v2,
           0, 0, 0);
}
```

在上面的代码序列中，“format”为消息格式的数组，“msgNumber”是格式数组的下标，而“logMag”是采用固定类型（第一个为串而其余六个为整数）的固定数目的参数（7个）的标准系统记录调用。

20

现在参见图2与3，为了比较生成记录输出的先有技术与当前策略，图2中示出先有技术策略而图3中示出当前策略。在先有技术方法中，利用通常的机构可获得通用调用100：作为操作系统设备的函数库，或对编程语言通用的语句。通用调用100具有若干参数，是否使用一个参数取决于语句的实例中所使用的其它参数。在本创造性方法中，通用调用100是通过定义成对具体的记录操作（诸如将记录数据放在队列上）类别具有特定的参数数目与/或相应类型的函数150间接访问的。函数定义包含适当的通用语法并将函数的参数适当的确定路

25

由到通用调用 100 的参数上。在程序执行时程序员的代码中使用带有其适当的参数组 140 的函数调用 130 来生成适当的记录事件。作为中介函数的结果，参在编译时间而不是在运行时间检验参数的数目与/或类型。

- 5 参见图 4, 通过将函数调用的路由确定为通过宏及自动化宏定义进程, 记录消息系统能提供可用于错误检验的否则是冗长的诊断设备。例如, 可将文件名与行号有条件地附加在记录消息上。(注意: 在发布编译命令时能指定编译器选项。通常编译器选项之一是为开发或生产版本编译。这是条件“有条件地”所指称的。换言之, 生成两组宏
- 10 定义。一组包含附加的诊断信息, 文件与行号, 而另一组不包含。然后用预编译器条件指示语句-if/then/else-包裹这两组宏定义。如果是为开发编译代码, 便使用带附加信息的组。如果是为生产版本编译代码, 便使用另一组, 因为用户对附加信息不感兴趣)。这可通过令
- 15 预处理器脚本在宏定义上附加文件名与行号的预编译器指示语句(诸如, #define LOG\_LoggingVersion(v0) (vlog3Event(LoggingVersion, v0, \_\_LINE\_\_, \_\_FILE\_\_))) 来完成。这无须程序员协助便能完成并为开发接通而为生产断开。这样, 宏脚本 210 分析消息目录 200 来生成包含诊断信息的宏定义。

- 20 对于熟悉本技术的人员显而易见本发明不限于上文中示例性实施例的细节, 而可以以其它特定形式实施本发明而不脱离其精神或主要属性。因此在所有方面都认为当前的实施例是说明性的而非限制性的, 本发明的范围是用所附权利要求而不是用上文中的描述指定的, 因此旨在将进入权利要求的等效物的含义与范围中的所有改变都包括在其中。

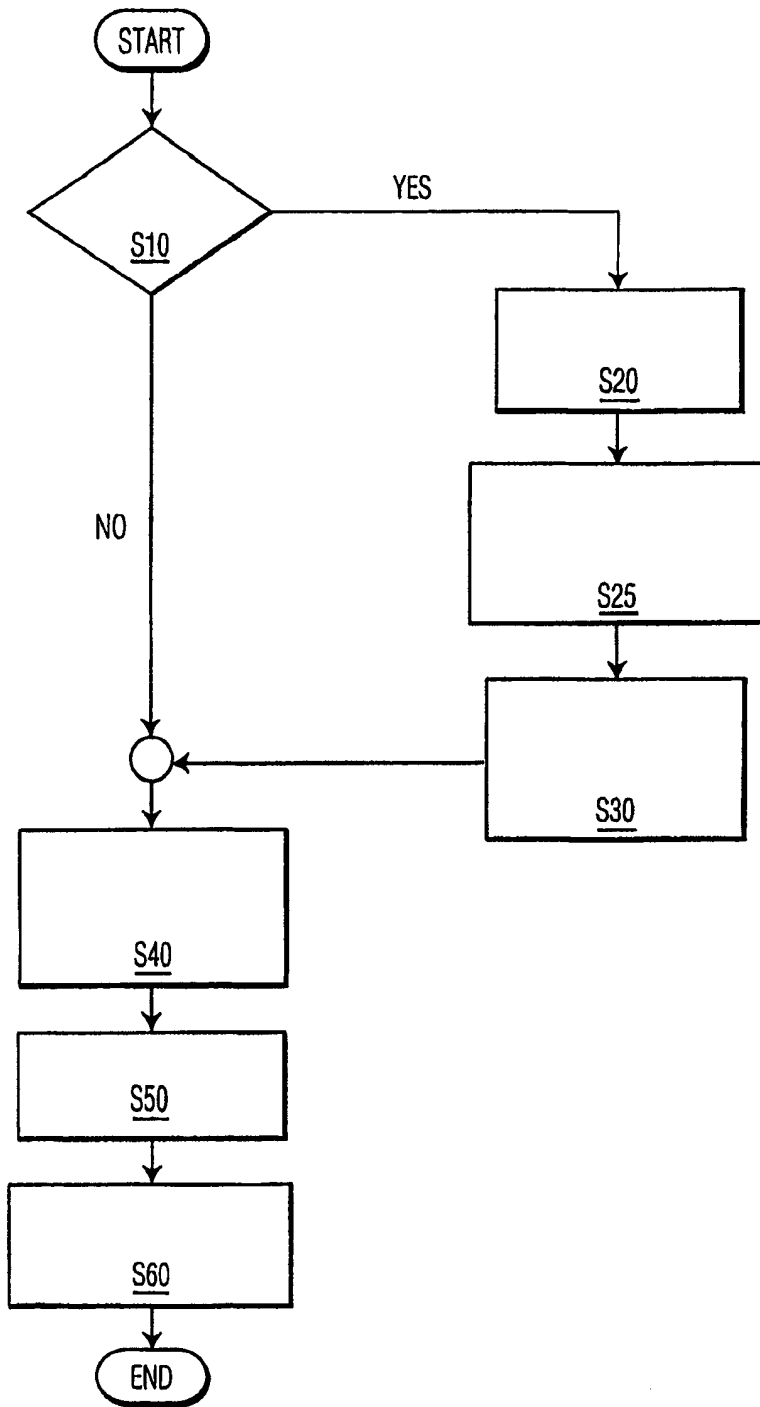


图 1

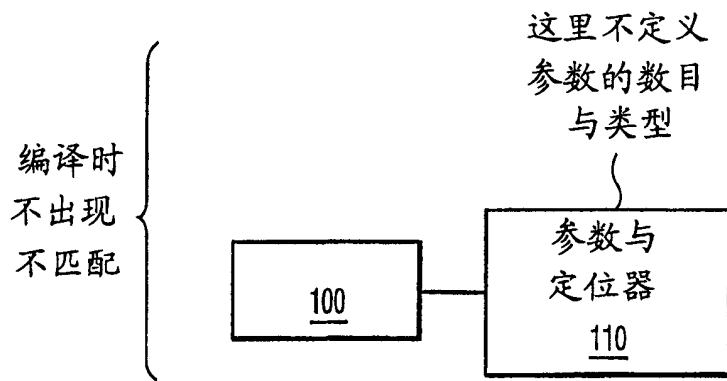


图 2

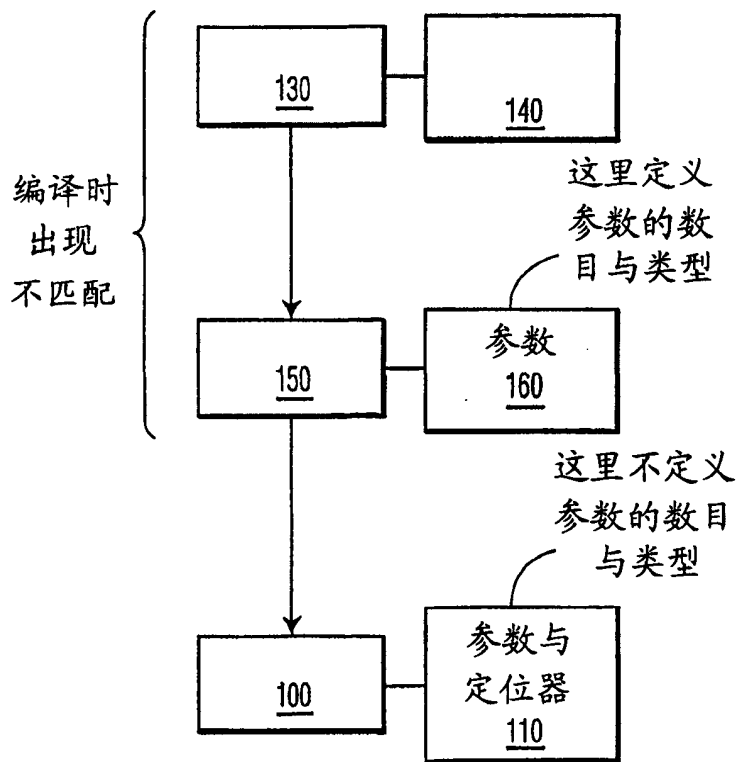


图 3

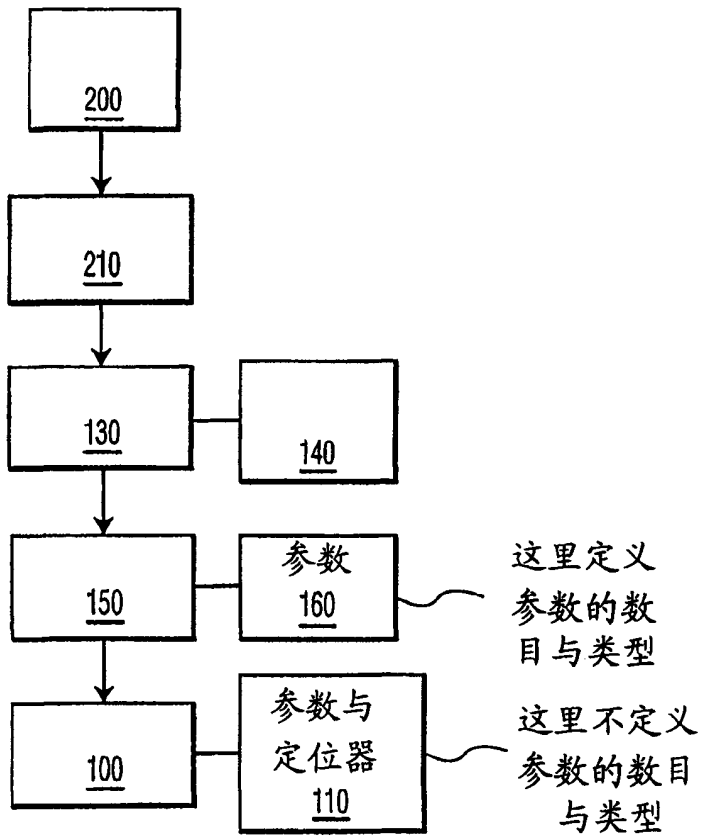


图 4